UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/592,368	06/12/2000	Edouard Bugnion	VMware5	5031

7590 09/09/2004

Jeffrey Pearce
34825 Sultan-Startup Rd
Sultan, WA 98294

EXAMINER	
SIDDIQI, MOHAMMAD A	

ART UNIT	PAPER NUMBER
2154	

DATE MAILED: 09/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/592,368	BUGNION, EDOUARD
	Examiner	Art Unit
	Mohammad A Siddiqi	2154

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 June 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,5,6 and 8-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,5,6,8-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1,5,6,8-24 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1,5,6,8-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yates et al. (6549959) (hereinafter Yates) in view of Krishnaswamy et al. (6,308,318) (hereinafter Krishnaswamy).

4. As per claim 1, Yates teaches the invention substantially as claimed:

- Yates teaches in a system in which a hardware target computer system (figure 1A, col 22, lines 15 – 46), which has a target instruction set architecture (ISA)(figure 1A-184, 194, col 22, lines 36-

46), executes a target instruction sequence corresponding to a source instruction sequence of a source system (figure 1A, col 22, lines 47-67), which has a source ISA and is running on the target computer system (figure 1a, col 27, lines 41-49) a method for handling exceptions (figure 3h, 3i, 3j, col 34, lines 45-65) comprising the following steps:

- converting the source instruction sequence into the target instruction sequence by binary translation (col 1, line 61) each instruction in the source instruction sequence being converted into a corresponding translated target instruction sequence which may consist of a single target instruction (col 1, lines 50-65);
- determining beginning and ending addresses of each source instruction and each corresponding translated target instruction (col 80, lines 21-52);
- generating a mapping between the beginning and ending addresses of each source instruction and its corresponding translated target instruction sequence (col 5, lines 8-14).
- executing the translated target instruction sequence (col 1, lines 60-65).
- sensing the presence of an exception (figure 3a, element 350 and 388,Col 21, lines 22-23) and determining whether each sensed

exception is synchronous or asynchronous (figure 3h, 3j, col 40, lines 24-39), a synchronous exception being defined as an exception resulting from attempted execution of a target instruction (figure 3j, col 40, lines 24-33) and an asynchronous exception being defined as an exception resulting from an event unrelated to the execution of a target instruction (figure 3h, 3j, col 38, lines 2-8).

- If the sensed exception is asynchronous (col 38, lines 5-8), resuming to completion the execution of the target instruction sequence in binary translation before handling the asynchronous exception (col 38, lines 1-8) and handling of each sensed exception until and no later than, completion of execution of the target instruction sequence corresponding to the current source instruction when the asynchronous exception is sensed (col 40, lines 24-65 and col 102, lines 1-9).

Yates fails to disclose specifically delaying handling of each asynchronous sensed exception, at any point in the translated target instruction, from the point in the translated target instruction sequence at which the asynchronous exception was sensed.

Krishnaswamy discloses delaying handling of each asynchronous sensed exception (col 3, lines 45-67), at any point in the translated target instruction (col 3, lines 45-67, col 4, lines 1-25), from the point in the

translated target instruction sequence at which the asynchronous exception was sensed (col 3, lines 45-55, col 4, lines 11-20, col 5, lines 10-17, col 6, lines 1-15).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to improve upon exception handler taught by Yates' provides a flexibility to have a single common strategy for processing all exceptions raised during the binary translation either by source or target ISA and execute exception handler on either source ISA or target ISA.

5. As per claims 5 and 19, Yates teaches a method in which synchronous exceptions are of either of two types, namely, transparent (figure 3h, 3j, col 34 lines 46-52) and non-transparent (it is implied if an exception is synchronous, and not transparent, then it must be nontransparent exception), a transparent exception being defined as an exception requiring processing action wholly within the target computer system (figure 3h, 3j, col 34 lines 46-52), and a non-transparent exception being defined as an exception requiring processing that alters a visible state of the source system, further including the following steps (figure 3h,3j, col 34 lines 46-52, it is implied if an exception is synchronous, and not transparent, then it must be nontransparent exception):

determining whether the sensed synchronous exception is transparent or nontransparent (figure 3a, 3l);

handling each transparent synchronous exception externally from the source system, the visible state of the source system thereby remaining unaltered (figure 3a – 3l); and

forwarding to the source system for processing each non-transparent synchronous exception (figure 3h, 3j, col 40, lines 24-39).

6. As per claim 6, Yates teaches a method of forwarding each non-transparent synchronous exception (figure eh, 3j, col 34 lines 46-52, it is implied if exception is synchronous and not transparent then it must be nontransparent) to the source system includes the step of converting the sensed exception into a simulated source exception in a source instruction stream, which is sensed by and interrupts the source system (figure 3a, 3l).

7. As per claim 8, Yates teaches a method determining a source instruction pointer as a predetermined function of the final target instruction pointer (figure 3j,col 40, lines 32-48);
forwarding and processing the sensed asynchronous exception (figure 3j,col 40, lines 32-48);

resuming execution at the location in the translation cache that corresponds to a current source instruction pointer (figure 3j,col 40, lines 32-39); and asynchronous exceptions thereby being processed only upon completion of execution of the translated target instructions corresponding to whole source instructions (col 41, lines 1-27).

8. As per claims 9 and 20, Yates teaches the exception (figure 3H) further includes the step of simulating execution of the remaining target instructions (col 18, lines 36 –67, col 19, lines 1-13).

Yates fails to disclose specifically delaying processing of the sensed asynchronous exception.

However, Krishnaswamy discloses delaying processing of the sensed asynchronous exception (col 3, lines 45-67, col 4, lines 1-24).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to improve upon exception handler taught by Yates' provides a flexibility to have a single common strategy for processing all exceptions raised during the binary translation either by source or target ISA and execute exception handler on either source ISA or target ISA.

9. As per claims 10 and 21, Yates teaches a method of processing exception further includes the step of single-stepping the execution of the remaining target instructions (col 31, lines 41-49).

Yates fails to disclose specifically delaying processing of the sensed asynchronous exception.

Krishnaswamy discloses delaying processing of sensed asynchronous

It would have been obvious to one of ordinary skill in the art at the time of invention was made to improve upon exception handler taught by Yates' provides a flexibility to have a single common strategy for processing all exceptions raised during the binary translation either by source or target ISA and execute exception handler on either source ISA or target ISA

10. As per claims 11 and 22, Yates teaches temporarily replacing with a trap generation instruction the initial target instructions in each of the translated target instruction sequences that correspond to target instruction sequences that possibly immediately follow the current target instruction sequence (fig 4b,col 58, lines 44-67);

resuming execution of the current target instruction sequence from the point at which the asynchronous exception was sensed (figure 3h, 3k, col 43, lines 3-40);

restoring each of the temporarily replaced instructions with their original content after completion of the processing of the sensed asynchronous exception (figure 3h, 3k, col 43, lines 3-40);

upon reaching the trap generation instruction, forwarding and processing the sensed asynchronous exception (figure 3h, 3k).

11. As per claim 12, Yates teaches temporarily replacing with a trap generation instruction each indirect branch instruction, each indirect branch instruction corresponding to a possible last instruction of the current target instruction sequence (fig 4b,col 58, lines 44-67);

resuming execution of the current target instruction sequence from the point at which the asynchronous exception was processed (figure 3h, 3k, col 43, lines 3-40);

restoring each of the temporarily replaced instructions with their original content (figure 3h, 3k, col 43, lines 3-40);

simulating the restored indirect branch instruction (col 56, lines 37-55).

12. As per claims 13 and 23, Yates teaches the source system is a virtual machine; a virtual machine monitor that is operationally installed between the virtual machine and the hardware target computer system, the virtual

machine thereby running on the virtual machine monitor (col 3, lines 13-30);

the steps of converting the source instruction sequence into the target instruction sequence by binary translation (col 1, lines 57-65), executing the translated target instruction sequence (col 11, lines 1 –11), sensing the presence of an exception (figure 3a-30), and application of the sensed exception (col 102, lines 1-9), are carried out by the virtual machine monitor (col 101, lines 17-32).

13. As per claims 14 and 24, Yates teaches source ISA is identical to the target ISA (col 1, lines 36-65).

14. As per claim 15, Yates teaches the invention substantially as claimed:

- Yates teaches in a system in which a hardware target computer system (col 1, lines 14-22), which has a target instruction set architecture (ISA) (col1, lines 36-46), executes a target instruction sequence corresponding to a source instruction sequence of a source system (col1, lines 11-13), which has a source ISA and is running on the target computer system (col 1, lines 13-25), a method for handling exceptions comprising the following steps:

- converting the source instruction sequence into the target instruction sequence by binary translation, each instruction in the source instruction sequence being converted into a corresponding translated target instruction sequence (col 1, lines 13-25, and col 1, lines 60-65);
- executing the translated target instruction sequence (col 1, lines 60-65;
- sensing the presence of an exception (figure 3a-350,Col 21, lines 22-23),
- each exception being of either of two types -- synchronous and asynchronous - a synchronous exception being defined as an exception resulting from attempted execution of a target instruction and an asynchronous exception is defined as an exception resulting from an event unrelated to the execution of a target instruction (figure 3h, 3j, col 40,lines 24-39),
- synchronous exceptions being of either of two types, namely, transparent (figure 3a-3f, col 34, lines 46-52) and non-transparent (it is implied if an exception is synchronous, and not transparent, must be nontransparent exception), a transparent exception being defined as an exception requiring processing action wholly within the target computer system (figure 3a-3f, col 34, lines 46-52), and a non-transparent exception being defined as an exception requiring

processing that alters a visible state of the source system(figure 3h, 3j, col 34, lines 24-39);

- determining whether each sensed exception is synchronous or asynchronous (figure 3a-3l);
- determining whether each sensed synchronous exception is transparent or non transparent (figure 3h, 3j, col 34 lines 46-52, it is implied if an exception is synchronous, and not transparent, then it must be nontransparent exception);
- upon sensing the presence of an asynchronous exception during execution of a current one of the translated target instruction sequences (figure 3h, 3j, col 40, lines 24-39), processing of the sensed asynchronous exception until completion of the remaining target instructions in the current translated target instruction sequence (col 102, lines 17-32);
- determining beginning and ending addresses of each source instruction and each corresponding translated target instruction (col 80, lines 21-52);
- generating a mapping between the beginning and ending addresses of each source instruction and its corresponding translated target instruction sequence (col 5, lines 8-23);

- handling each transparent synchronous exception externally from the source system, the visible state of the source system thereby remaining unaltered (figure 3h, 3j, col 40, lines 24-39);
- forwarding to the source system for processing each non-transparent

synchronous exception (figure 3a –30);

- determining a source instruction pointer as a predetermined function of the final target instruction pointer (figure 1a-602, col 22, lines 56-67);

forwarding and processing each sensed asynchronous exception (figure 3h, 3j, col 40, lines 24-39);

- resuming execution at the location in the translation cache that corresponds to a current source instruction pointer (figure 3h, 3k, col 43, lines 3-40), asynchronous exceptions thereby being processed only upon completion of execution of the translated target instructions corresponding to whole source instructions (figure 3h, 3k, col 43, lines 3-40);
- each asynchronous sensed exception until (col 40, lines 24-40), lines, and no later than, completion of execution of the target instruction sequence corresponding to the current source instruction when the asynchronous exception is sensed (col 40, lines 40-65).

in which:

- the source system is a virtual machine (col1, lines 35-65);
- a virtual machine monitor that is operationally installed between the virtual machine and the hardware target computer system, the virtual machine thereby running on the virtual machine monitor (figure 1a, col 3, lines 13-30); and
- the steps of converting the source instruction sequence into the target instruction sequence by binary translation (col 1, lines 57-65), executing the translated target instruction sequence (col 11, lines 1-11), sensing the presence of an exception (figure 3a-30), and the sensed exception (col 102, lines 1-9), are carried out by the virtual machine monitor (col 101, lines 17-32).

Yates fails to disclose specifically delaying handling of each asynchronous sensed exception, at any point in the translated target instruction, that is, from the point in the translated target instruction sequence at which the asynchronous exception was sensed.

Krishnaswamy discloses delaying handling of each asynchronous sensed exception (col 3, lines 45-67), at any point in the translated target instruction, that is, from the point in the translated target instruction sequence at which the asynchronous exception was sensed (col 3, lines 45-55, col 4, lines 11-20, col 5, lines 10-17, col 6, lines 1-15).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to improve upon exception handler taught by Yates' provides a flexibility to have a single common strategy for processing all exceptions raised during the binary translation either by source or target ISA and execute exception handler on either source ISA or target ISA.

15. As per independent claim 16, Yates teaches a hardware target computer system that has a target instruction set architecture (ISA) and executes a target sequence (col 1, lines 14-25);

- a source system that has a source ISA and a source instruction sequence (col 1, lines 35-65);
- an intermediate software layer forming an interface between the source system and the hardware target computer system (col 22, lines 33-67 and col 27, lines 1-50);
- a binary translator included within the intermediate software layer (col 22, lines 33-67) converting the source instruction sequence into the target instruction sequence by binary translation (col 23, lines 19-41), each instruction in the source instruction sequence being converted into a corresponding translated target instruction sequence which may consist of a single target instruction (col 23, lines 19-41 and col 27, lines 1-50); and

- an exception handler included within the intermediate software layer and sensing the presence of an exception and determining whether the exception is synchronous or asynchronous (col 40, lines 24-39);
- a resume component within the intermediate software layer for resuming to completion the execution of the target instruction sequence in binary translation if the sensed exception is asynchronous 9col 40, lines 24-39), before handling of the asynchronous exception (col 41, lines 18-25) and
- the exception handler further process asynchronous sensed exception until (col 40, lines 24-40), and no later than, completion of execution of the target instruction sequence corresponding to the current source instruction when the asynchronous exception is sensed (col 40, lines 40-65).

Yates fails to disclose specifically delaying handling of each asynchronous sensed exception, at any point in the translated target instruction sequence, that is, from the point in the translated target instruction sequence at which the asynchronous exception was sensed.

Krishnaswamy discloses delaying handling of each asynchronous sensed exception (col 3, lines 45-67), at any point in the translated target instruction sequence, that is, from the point in the translated target

instruction sequence at which the asynchronous exception was sensed (col 3, lines 45-55, col 4, lines 11-20, col 5, lines 10-17, col 6, lines 1-15).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to improve upon exception handler taught by Yates' provides a flexibility to have a single common strategy for processing all exceptions raised during the binary translation either by source or target ISA and execute exception handler on either source ISA or target ISA.

16. As per claim 17, it is rejected for similar reasons as stated in claim 1, above.

17. As per claim 18, it is rejected for similar reasons as stated in claim 1.

Response to Arguments

1. Applicant's arguments with respect to claims 1, 16 and 17 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

2. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mohammad A Siddiqi whose telephone number is (703) 305-0353. The examiner can normally be reached on Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (703)305-8498.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MAS